Amendments to the Drawings:

Included in the amendment are an "Annotated Sheet Showing Changes" and a "Replacement Sheet" for Figs. 4C, 8, and 9.

In Fig. 4C the equations in Syntax/Operation block 460 have been modified to correspond with the specification and to correct three typos. Two of the typos concern a parenthesis inadvertently included in an equation and the third typo concerns the labeling of the 16th instruction. The equation:

(V[01]+VIMOFFS) + 1)[UnitVIM]← 2nd Instruction following LV2

and

(V[01]+VIMOFFS) + InstrCnt)[UnitVIM]←(InstrCnt)th Instruction following LV2

have been amended to:

(V[01]+VIMOFFS) + 1)[UnitVIM]← 2nd Instruction following LV2

and

(V[01]+VIMOFFS) + InstrCnt)[UnitVIM]←(InstrCnt+ 1)th Instruction following LV2

to correctly correspond to the text immediately below the last equation which correctly states: "InstrCnt is a binary coded number, 0 thru F, that represents from 1 to 16 instructions that can be loaded into up to 16 consecutive UnitVIM locations." For example, for the 16th instruction to be loaded, InstrCnt would be set to a maximum setting of F, a binary setting of all 1's in the four bit InstrCnt bit field of bits 18-21 in LV2 instruction 450 Fig. 4C, and the equation would then indicate, by substituting F for InstrCnt, as follows:

(V[01]+VIMOFFS + F)[UnitVIM]←(F + 1)th Instruction following LV2

which equivalently in decimal is:

(V[01]+VIMOFFS + 15)[UnitVIM]←(16)th Instruction following LV2

Consequently, with (V[01]+VIMOFFS)=b as the start address where the first instruction is loaded, sixteen VIM addresses b to b+15 are loaded, where b+15 is the 16th instruction loaded.

In Fig. 8, high address label A-1 on the ALU VIM 903 is amended to C-1 and high address label B-1 on the adjacent MAU VIM is amended to D-1 to be consistent with partitioned VIM section high address labels of Fig. 6 ALU instruction VIM 624 and MAU instruction VIM 626, respectively. The capacities of the separate VIM sections are indicated by the 0-(A-1) store VIM 620, 0-(B-1) load VIM 622, 0-(C-1) ALU VIM 624, 0-(D-1) MAU VIM 626, and 0-(E-1) DSU VIM 628. Each separate VIM section as shown in Figs. 6 and 8 may have a different capacity depending upon application and implementation requirements.

In Fig. 9, high address label B-1 on the ALU VIM 903 is amended to C-1 to be consistent with partitioned VIM section high address labels of Fig. 6 ALU instruction VIM 624 and newly amended Fig. 8 ALU VIM 903. In addition, circular label d 937 is amended to c 937 to be consistent with the similar label in Fig. 8 associated with the VIM ALU 903.

Annotated Sheet Showing Changes

SERIAL NO.: ~~09/747,056~~ 10/758,548
PETER H. PRIEST (919-806-1600)

FIG. 4C

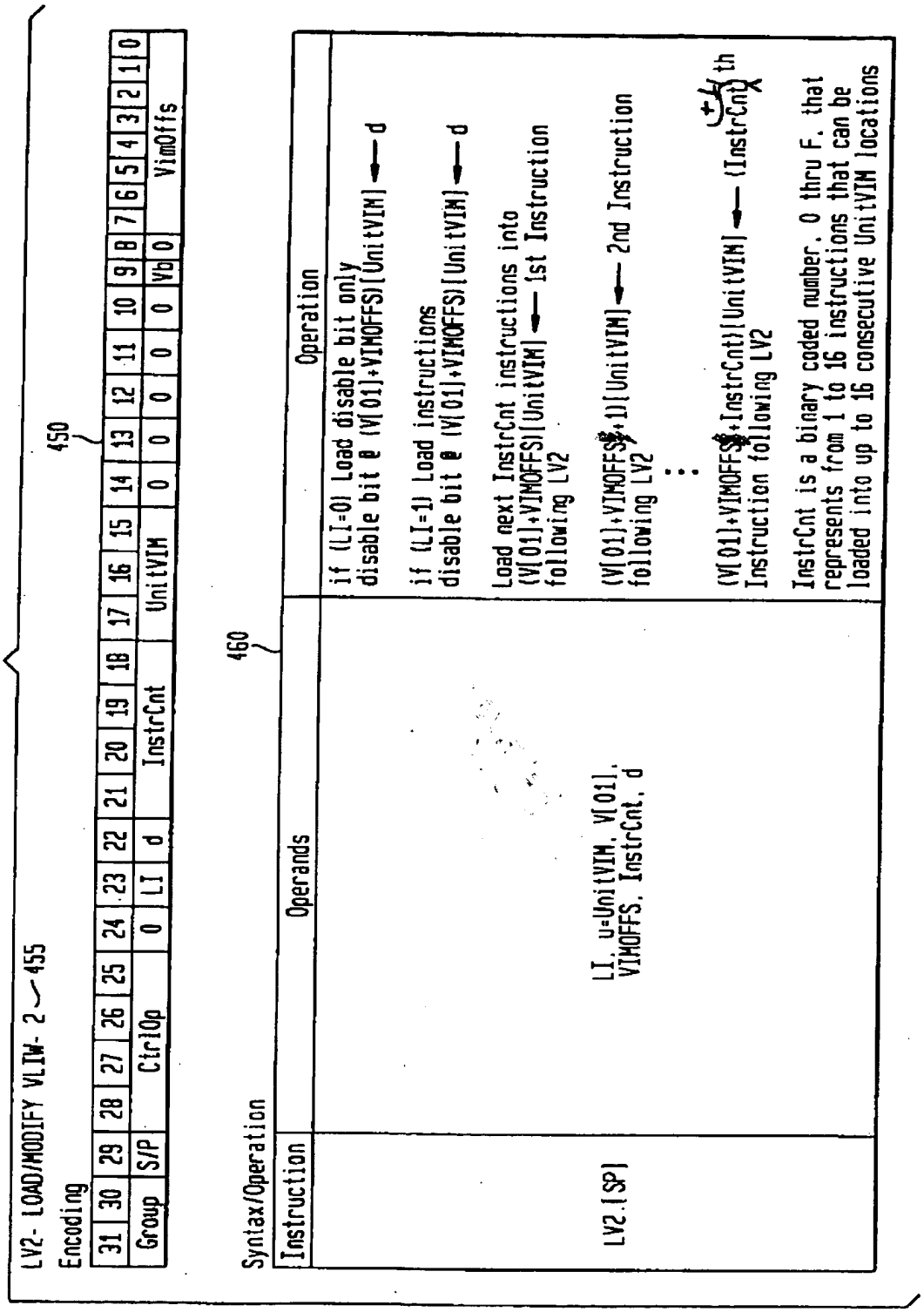
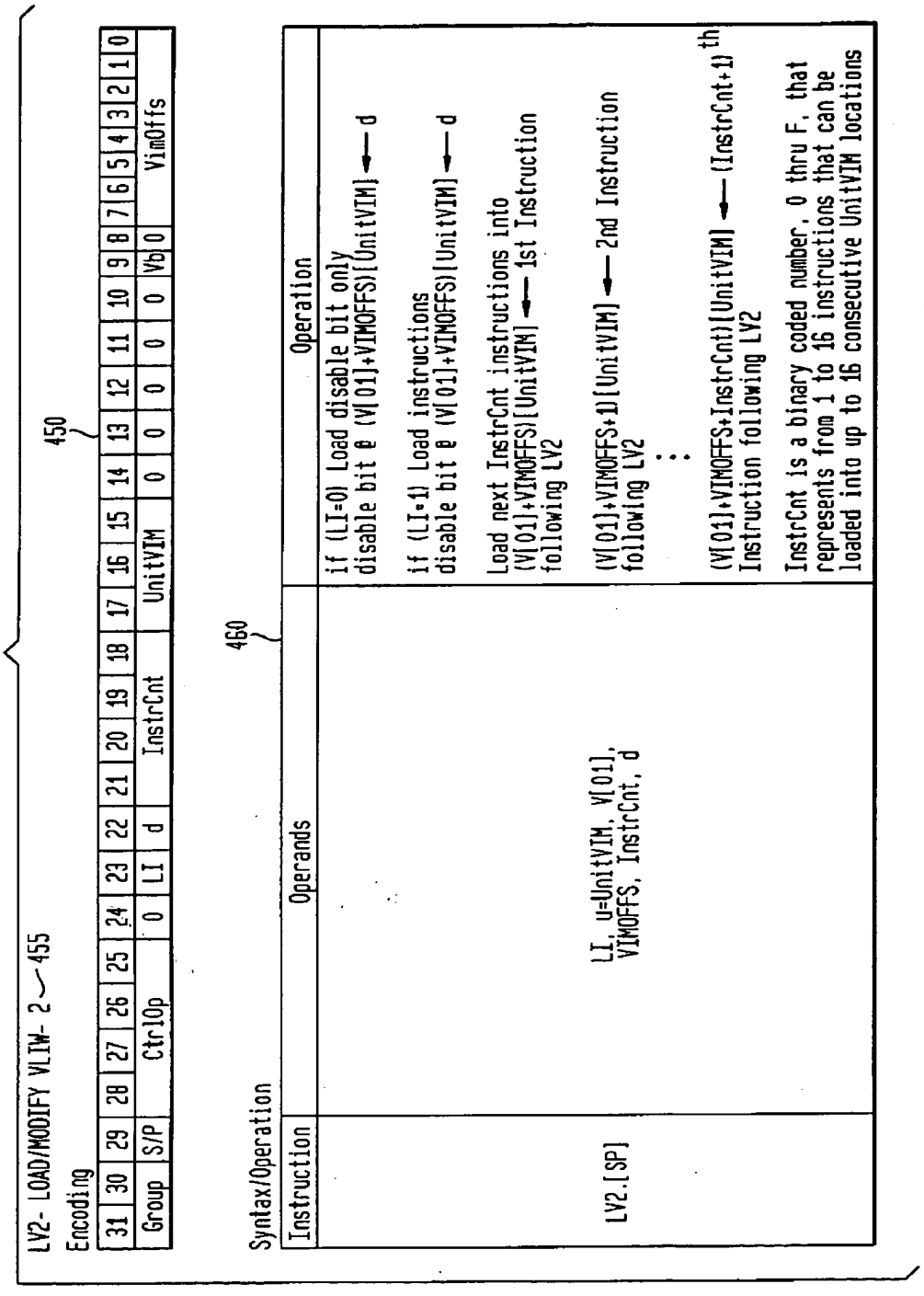


FIG. 4C


LV2- LOAD/MODIFY VLIW- 2
455
450
Encoding
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group S/P Ctrl0p 0 LI d InstrCnt UnitVIM 0 0 0 0 0 Vb 0 VimOffs
460
Syntax/Operation
Instruction
Operands
Operation
LV2.[SP]
LI, u=UnitVIM, V[01], VIMOFFS, InstrCnt, d
if (LI=0) Load disable bit only
disable bit @ (V[01]+VIMOFFS)[UnitVIM] ← d
if (LI=1) Load instructions
disable bit @ (V[01]+VIMOFFS)[UnitVIM] ← d
Load next InstrCnt instructions into
(V[01]+VIMOFFS)[UnitVIM] ← 1st Instruction
following LV2
(V[01]+VIMOFFS+1)[UnitVIM] ← 2nd Instruction
following LV2
(V[01]+VIMOFFS+InstrCnt)[UnitVIM] ← (InstrCnt+1)th
Instruction following LV2
InstrCnt is a binary coded number, 0 thru F, that
represents from 1 to 16 instructions that can be
loaded into up to 16 consecutive UnitVIM locations

SERIAL NO.: ~~09/747,056~~ 10/753,348
PETER H. PRIEST (919-806-1600)

Annotated Sheet Showing Changes

11/12

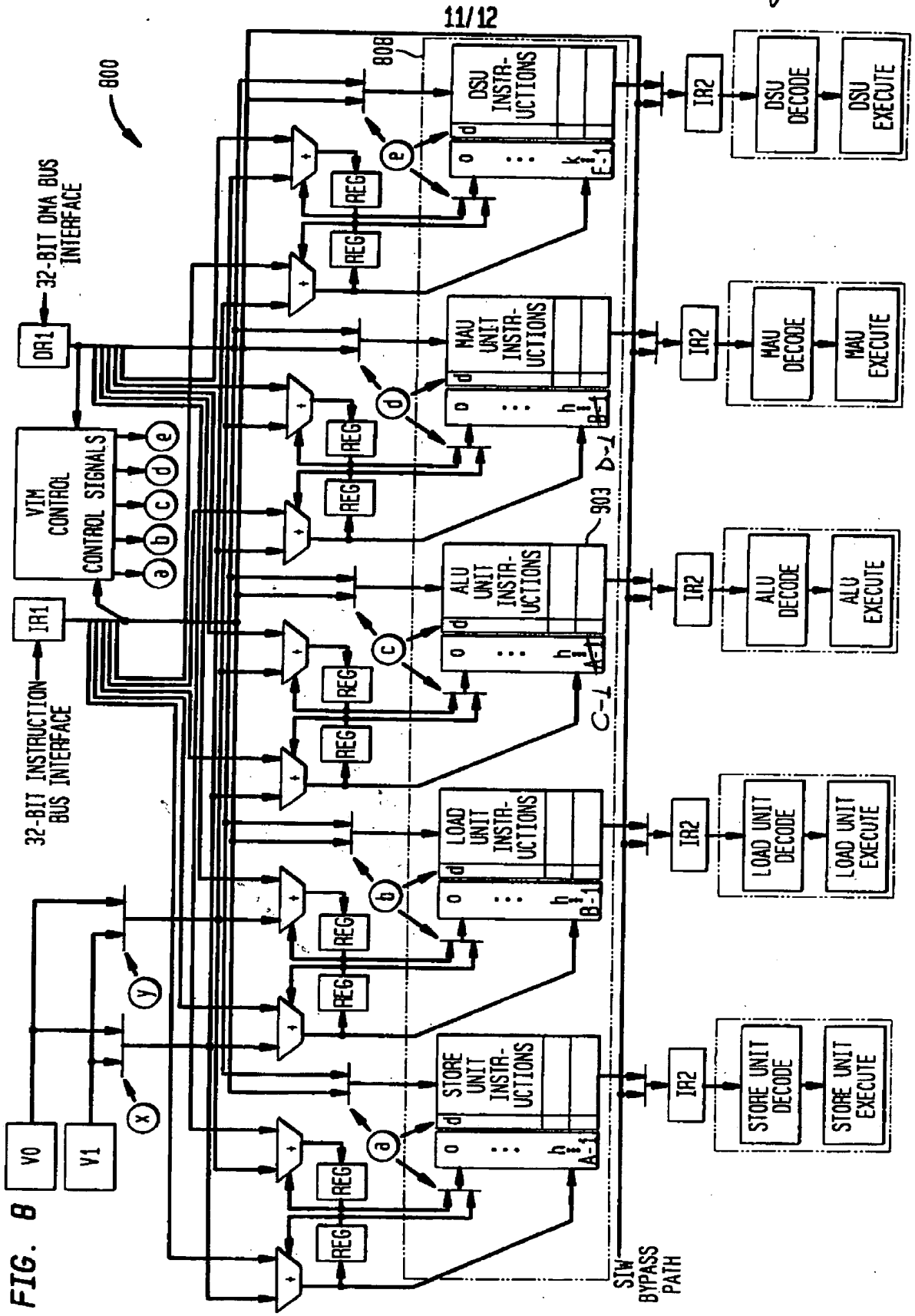


FIG. 8

SERIAL NO.: 10/758,348
PETER H. PRIEST (919-806-1600)

11/12

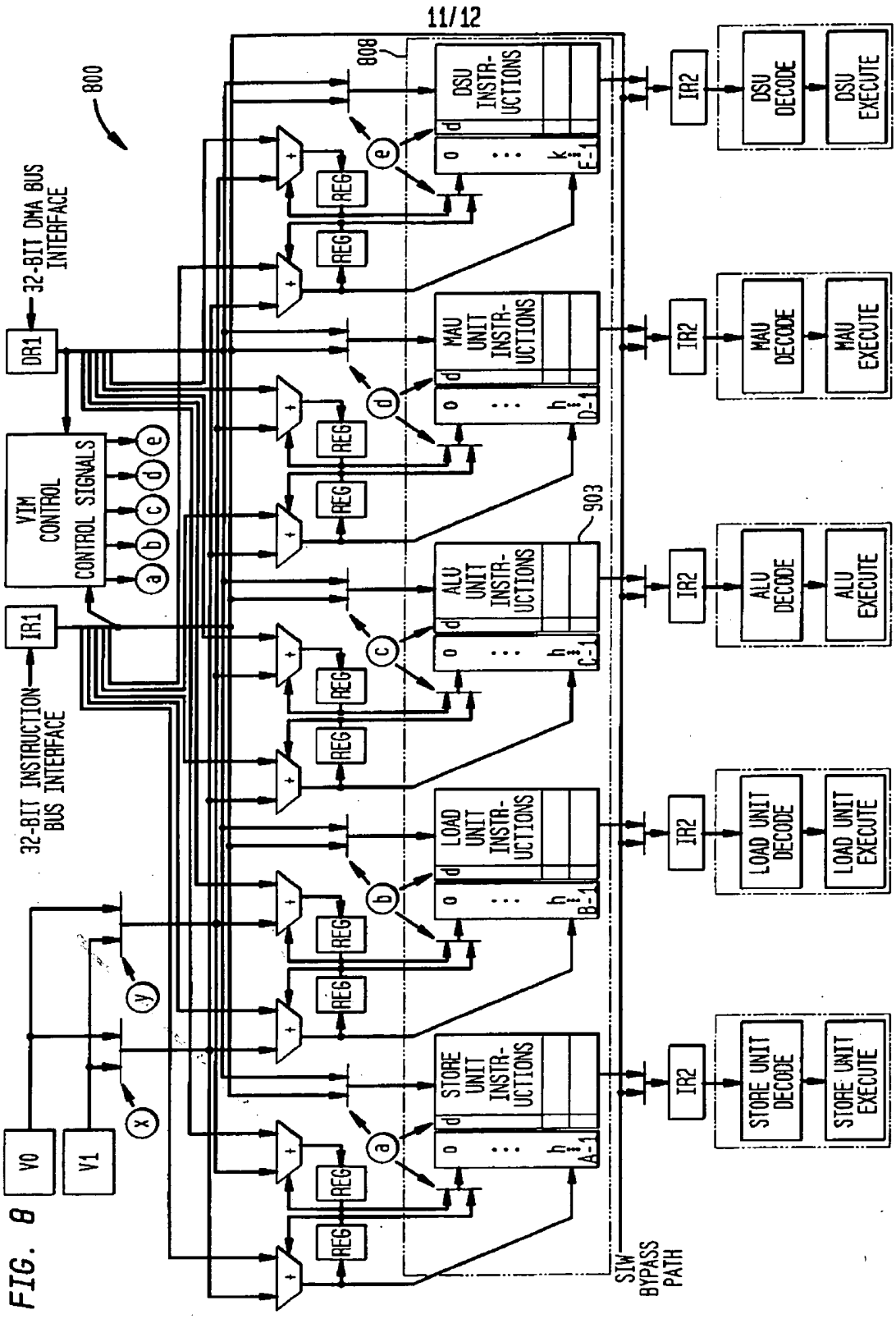


FIG. 8

SERIAL NO.: ~~09/747,056~~ 10/758,348
PETER H. PRIEST (919-806-1600)

12/12

Annotated Sheet Showing Changes

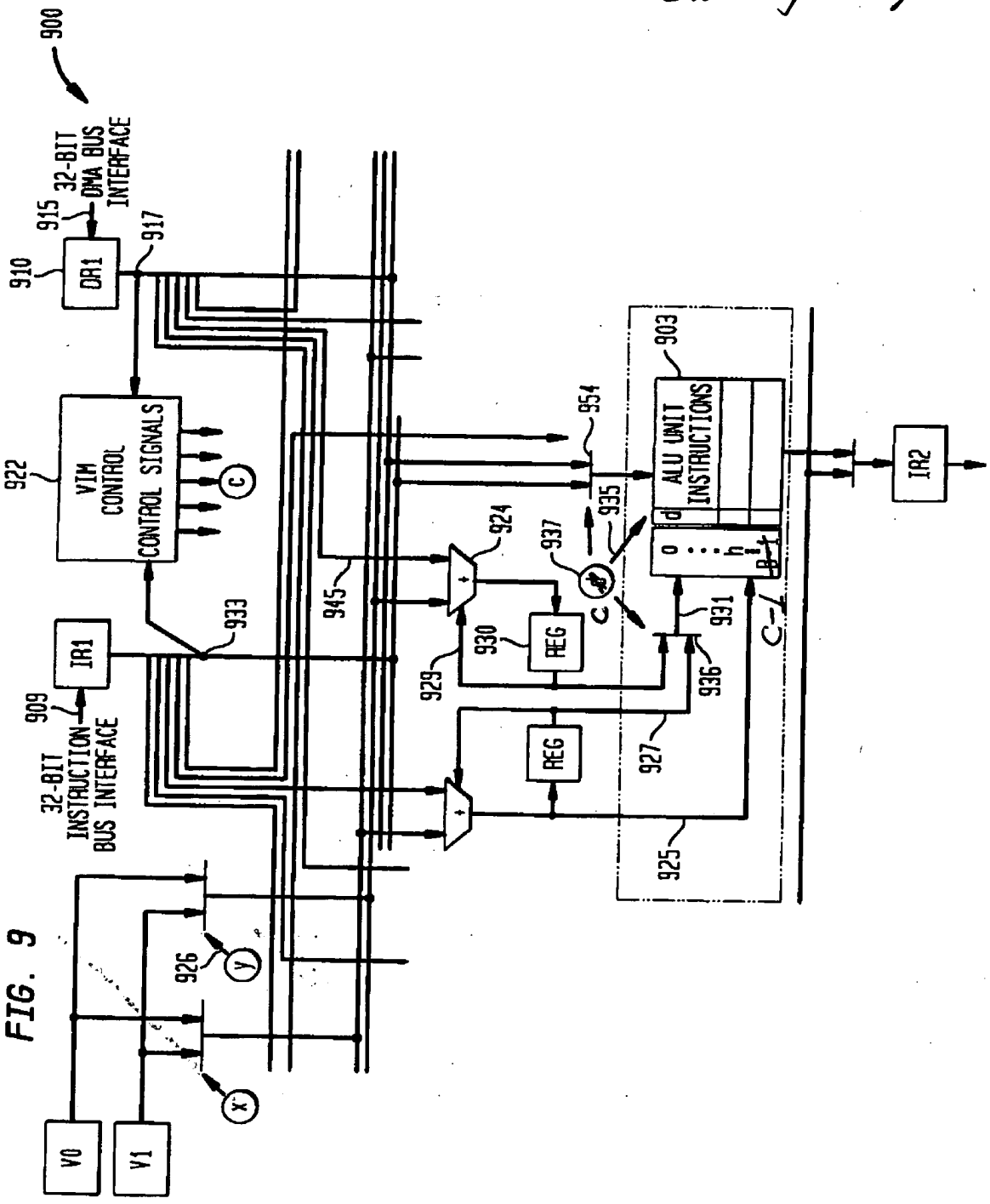


FIG. 9

SERIAL NO.: 10/758,348
PETER H. PRIEST (919-806-1600)

12/12

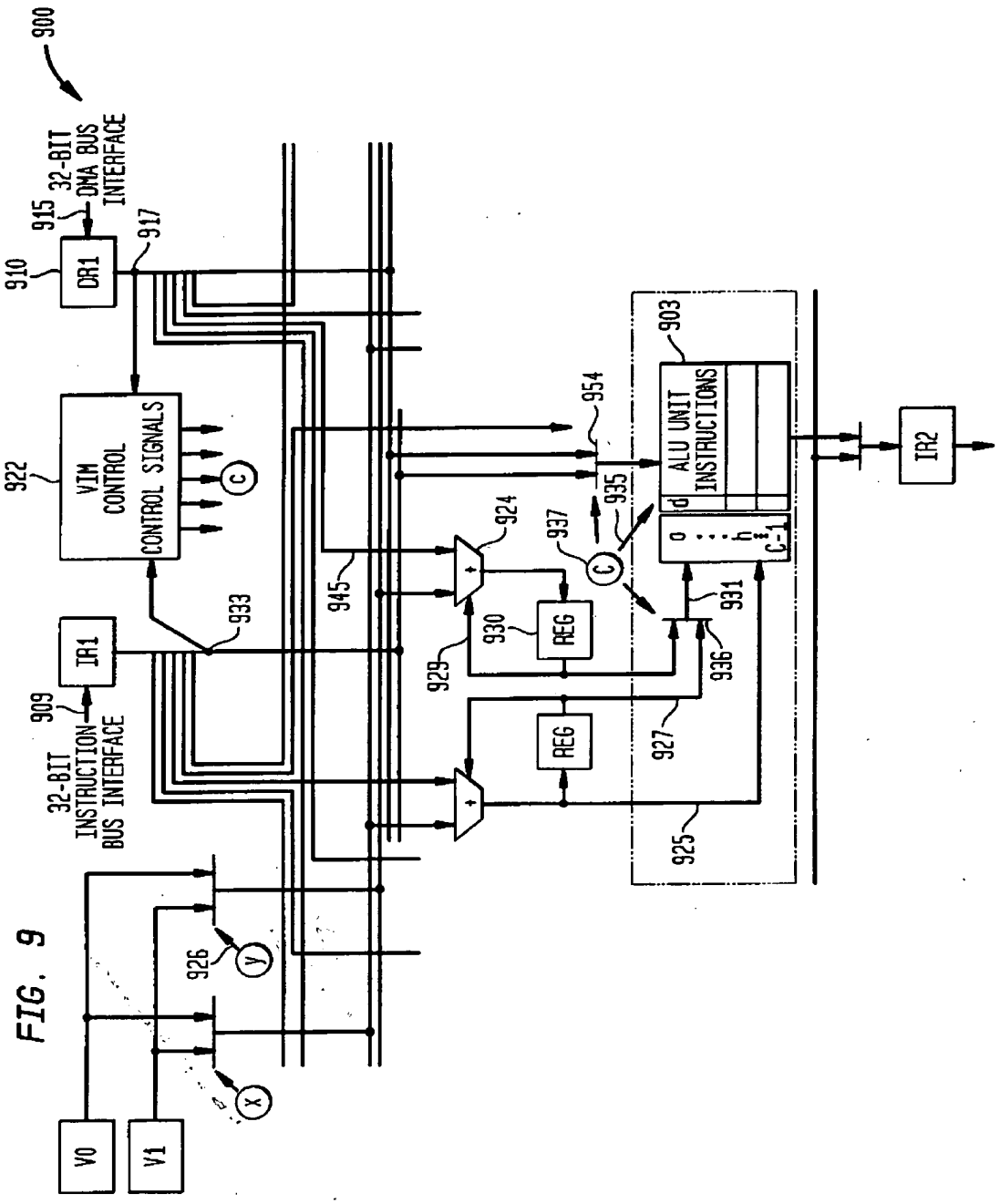


FIG. 9